Exploring Methods for Finding Maxima Using Bayesian Optimisation

Rebekah Fearnhead
Supervisor: Daniel Dodd

STOR-i, Lancaster University

August 26, 2022

Contents

Introduction

Introduction

- The global optimisation of black-box functions which are expensive and potentially gradient-free is an important problem in industry, for example in training self driving cars.
- Bayesian optimisation is an approach which has been shown to obtain better results, with fewer evaluations, compared to other methods such as random-search based methods.
- The general idea is to construct a probabilistic model of the objective function which can then be used to sequentially decide where to evaluate it next.
- This model can then be improved using a surrogate function which adds extra noise to the sampled data and these two methods will be compared.

Bayesian Optimisation

1. To perform Bayesian Optimisation, a prior is placed over the function to capture any beliefs about the behaviour of the function.
2. After data points are observed, the prior can be updated to form a posterior distribution for the function.
3. This posterior distribution, along with an acquisition function is then used to decide where to sample next and then this process is repeated.

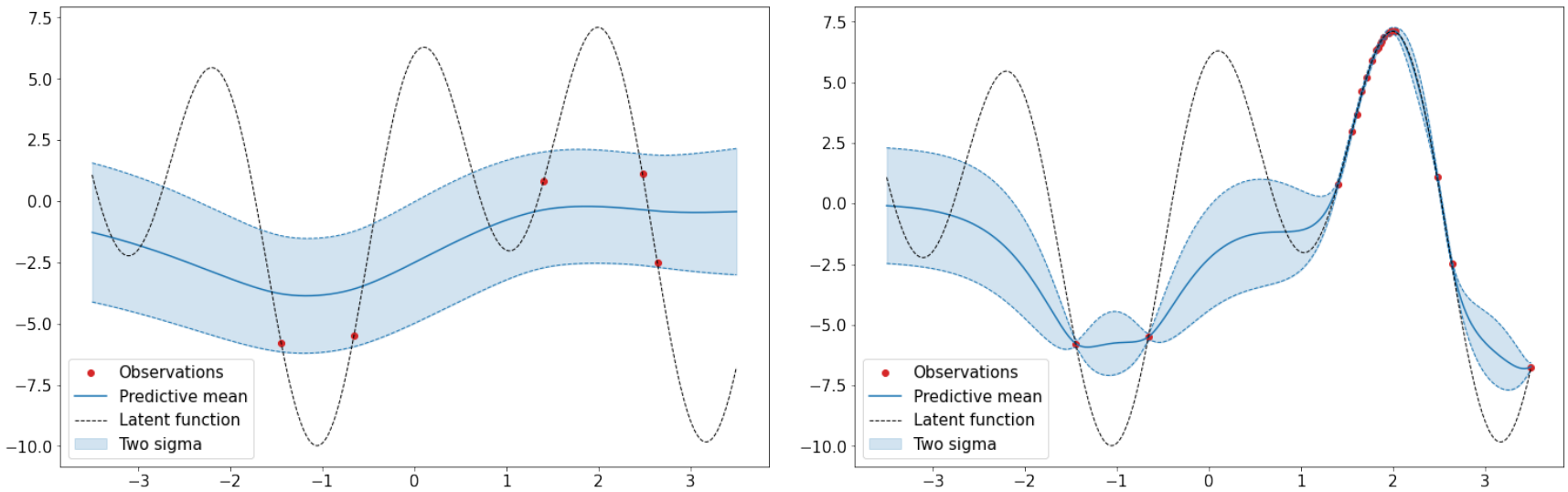


Figure: A graph of the predicted distribution of a function (blue) and two standard deviations of the prediction, given the observed data points (red), compared to the true function (black) after observing 5 and 20 data points respectively.

Acquisition Functions

Acquisition Functions

- Acquisition functions are used to decide the most beneficial places to sample to find the maximum (or minimum) of a function.
- Different types of functions place more importance on sampling points in one of two different ways to improve the search for the optimum:
  - Exploration: sampling points where there is a high uncertainty of the true value.
  - Exploitation: sampling points close to where the model prediction is already high.

Improvement Based Policies

- These focus on favouring points that are likely to improve on the current maximum value, τ.

Probability of Improvement

$$\alpha_{PI}(\mathrm{x}; \mathcal{D}_n) := \mathbb{P}[v > \tau] = \Phi\left(\frac{\mu_n(\mathrm{x}) - \tau}{\sigma_n(\mathrm{x})}\right). \tag{1}$$

Expected Improvement

$$\alpha_{EI}(\mathrm{x}; \mathcal{D}_n) := (\mu_n(\mathrm{x}) - \tau)\Phi\left(\frac{\mu_n(\mathrm{x}) - \tau}{\sigma_n(\mathrm{x})}\right) + \sigma_n(\mathrm{x})\phi\left(\frac{\mu_n(\mathrm{x}) - \tau}{\sigma_n(\mathrm{x})}\right). \tag{2}$$

Optimistic Policies

- These aim to be optimistic by looking at the upper confidence bounds for the predicted values of the data points that can be sampled.
- The hyper-parameter, β_n can be adjusted, with higher values placing more weight on exploration.

Upper Confidence Bound

$$\alpha_{UCB}(\mathrm{x}; \mathcal{D}_n) := \mu_n(\mathrm{x}) + \beta_n \sigma_n(\mathrm{x}). \tag{3}$$

Problems with Acquisition Functions

- Traditional acquisition functions are often trapped sampling a small area after locating a local optimum.
- This is caused by the greater importance placed on exploitation rather than exploration.
- Surrogate functions aim to resolve this by increasing the posterior variance which encourages exploration.

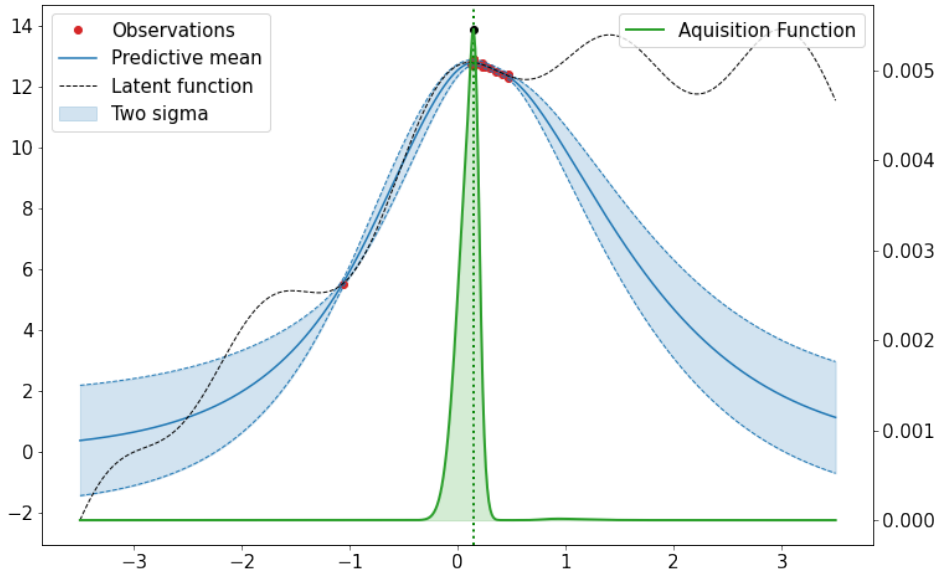


Figure: A graph of the expected improvement acquisition function getting trapped at a local maximum of the function being optimised.

Surrogate Functions

Surrogate Functions

- Surrogate functions improve the performance of acquisition functions by adding noise to the model which increases the posterior variance to encourage exploration.
- This can be shown as

$$f(\mathrm{x}) = g(\mathrm{x}, \mathrm{h}), \qquad g \sim \mathcal{GP}, \qquad \mathrm{h} \sim \mathcal{N}(0, \sigma_h), \tag{4}$$

  where g is a well behaved function following a GP prior distribution.
- This therefore adds non-linear interactions between a random variable, h and the sampled values of x to the predicted function distribution.

Comparing Methods

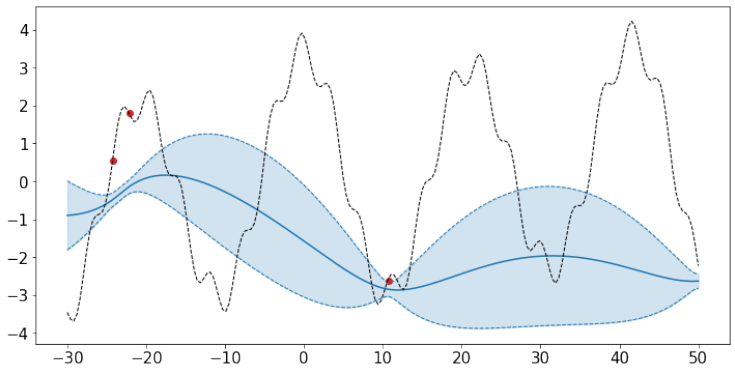
Simulation Setup

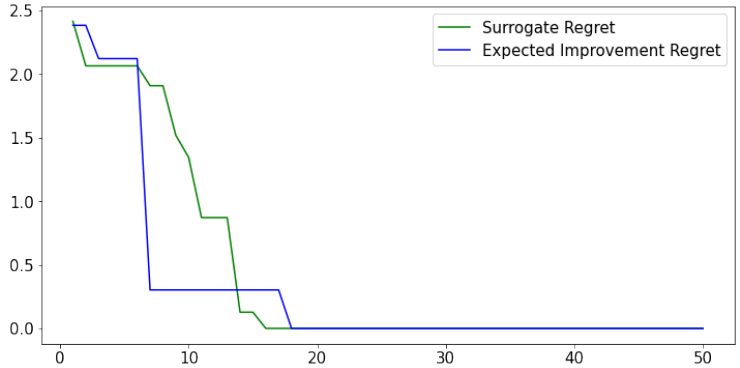
- The difference in performance of the use of a surrogate function before the acquisition function compared to just using an acquisition function can be can be shown by simulating the two methods on different functions.
- This simulation is done in Python using the GPJax module.
- For each simulation, the starting position is the same for expected improvement both with and without using a surrogate function, with the same data point values given.

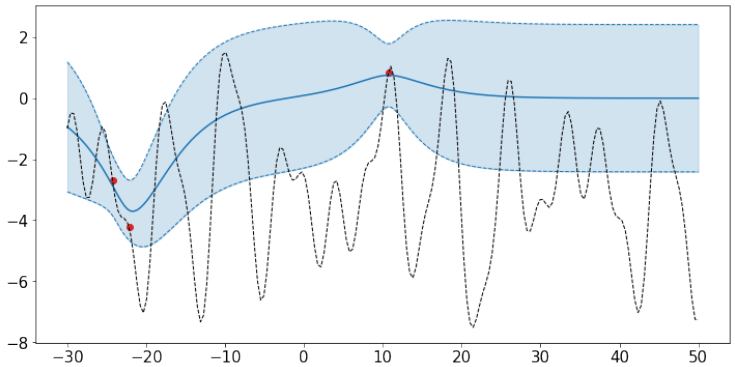
Simulation Setup

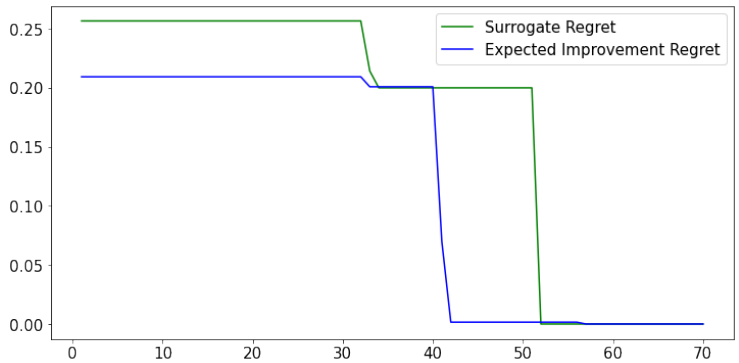
- The two methods can then be compared by plotting the regret of both of the predicted functions after each new data point is added to the model by the iteration of Bayesian optimisation.
- Regret is the difference between the highest value found by the sampling and the true global maximum.
- This means that the quicker this gets to zero, the more efficient the method is.

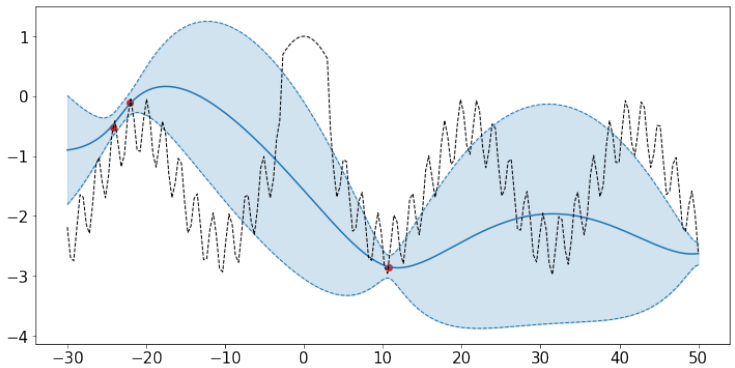
Results

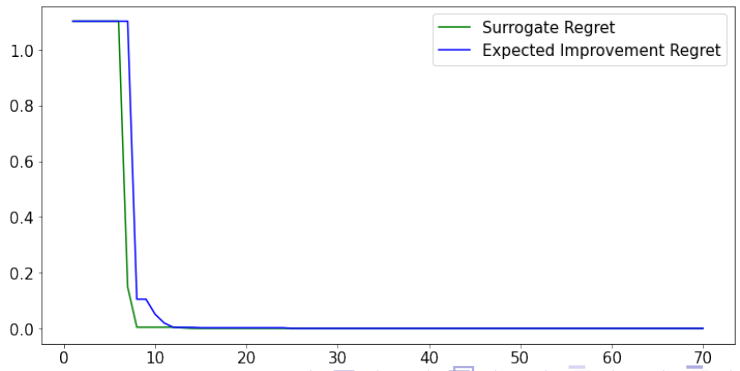



Surrogate Regret
Expected Improvement Regret




Surrogate Regret
Expected Improvement Regret




Surrogate Regret
Expected Improvement Regret

Observations

- From the tested functions, it can be seen that for the smooth function and the step function, using a surrogate function before applying Expected Improvement outperformed just using acquisition function.
- However, for the function with lots of local maxima, the acquisition function performs better. Even though the surrogate model gets a regret value of 0 first, using expected improvement gets close to the maximum quicker. This could be because the function already is quite noisy and unpredictable so the posterior variance is already large.

Further Investigations

Further Investigations

- Firstly the simulations on the three functions could be repeated with different starting points given. This would mean that it could be seen if the results show the improvement with using surrogate functions no matter the staring points.
- Also, the affects of changing the value of σ_h could be investigated by comparing the regret functions produced when running the simulations for different values on the same functions.

Thank you for listening

Any questions?

References

Bodin, E., Kaiser, M., Kazlauskaite, I., Dai, Z., Campbell, N., & Ek, C. H. (2020, November). Modulating surrogates for Bayesian Optimization. In *International Conference on Machine Learning* (pp. 970-979). PMLR.

Rasmussen, & Williams, Christopher K. I. (2006). *Gaussian processes for machine learning.* MIT Press.

Shahriari, Swersky, K., Wang, Z., Adams, R. P., & de Freitas, N. (2016). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE, 104*(1), 148–175.